

# Plain English Programming — Is a Picture Worth 1000 Words?

gerryrzeppa

Uncategorized

May 2, 2018

4 Minutes

It's been said that a picture is worth a thousand words. Let's find out.

We start with a picture of a famous person. In this case, I chose a photo of the famous athiest Richard Dawkins:



Then using this type definition...

A block is a thing with a box and a brightness.

...and this routine...

To create a list of brightness blocks from a picture:

Center the picture on the screen.

Draw the picture.

Put the picture's left-top into a spot.

Loop.

Allocate memory for a block. Append the block to the list.

Put the spot and the spot plus 12 pixels into the block's box.

Put the block's average brightness into the block's brightness.

Add 12 pixels to the spot's left. If the spot's left is less than the picture's right, repeat.

Put the picture's left into the spot's left.

Add 12 pixels to the spot's top. If the spot's top is less than the picture's bottom, repeat.

...I divided the photo into 12 pixel by 12 pixel blocks, calculated the average brightness of each block, and saved the blocks on a list. This is the function-style routine that calculates the average brightness for each block:

To put a block's average brightness into a brightness:

Put the block's left-top into a spot.

Loop.

Get a color given the spot.

Add the color's brightness to a total brightness. Add 1 to a count.

Add 1 pixel to the spot's left. If the spot is in the block's box, repeat.

Put the block's left into the spot's left.

Add 1 pixel to the spot's top. If the spot is in the block's box, repeat.

Put the total divided by the count into the brightness.

At this point, at least in my mind, the photo now looked something like this (imaginary grids in my head are always sky blue):





Then I picked a bold, monospaced font that had uppercase letters approximately 12 pixels square, and chose an appropriately ironic sentence (with the spaces removed so the resulting picture would have a uniform “texture”). Finally, I drew the sentence over and over, one letter in each block in the shade of gray indicated by the block’s brightness. This is the routine:

To draw a list of brightness blocks given a string and a font:

Get a letter from the string (wrapping at the end).

## Get a block from the list.

If the block is nil, break.

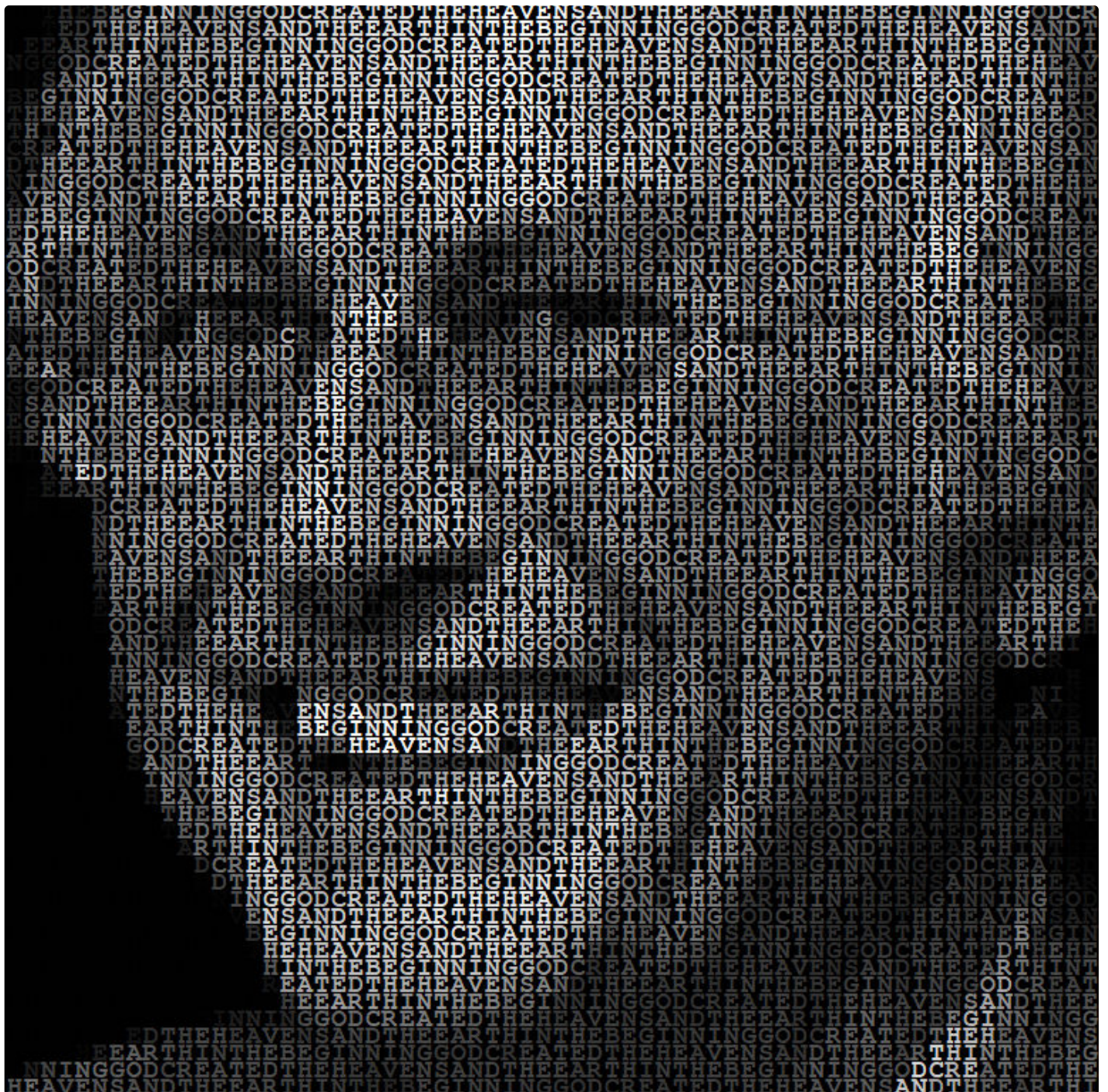
Put `o` and `o` and the block's brightness into a color.

Draw the letter in the center of the block's box with the color and the font.

Repeat.

Refresh the screen.

And this is the result I got on the screen:





The picture is 8 inches square. Plain English uses 96 pixels per inch, and our blocks are 12 pixels square, so there are 64 blocks (or letters) per square inch; 4096 in all. The average length of a word in English is 5 letters, so it looks like, in this case, a picture is only worth about 819 words!

This is the entire program in case you want to type it in yourself and run it using our Plain English development system ([www.osmosian.com/cal-4700.zip](http://www.osmosian.com/cal-4700.zip)) with any picture and string of your choice:

A block is a thing with a box and a brightness.

A list is some blocks.

To run:

Start up.

Fetch a picture from “c:\g4g articles\1000 words\dawkins.png”.

Create a list of brightness blocks from the picture.

Clear the screen.

Put “courier new bold” and 1/4 inch into a font.

Draw the list of brightness blocks using

“INTHEBEGINNINGGODCREATEDTHEHEAVENSANDTHEEARTH” and the font.

Wait for the escape key.

Destroy the picture.

Destroy the list.

Shut down.

To draw a list of brightness blocks given a string and a font:

Get a letter from the string (wrapping at the end).

Get a block from the list.

If the block is nil, break.

Put o and o and the block’s brightness into a color.

Draw the letter in the center of the block’s box with the color and the font.

Repeat.

Refresh the screen.

The byte pointer is a byte pointer.

To get a letter from a string (wrapping at the end):

If the byte pointer is not nil, add 1 to the byte pointer.

If the byte pointer is greater than the string’s last

put the string’s first into the byte pointer.

If the byte pointer is nil, put the string’s first into the byte pointer.

Put the byte pointer’s target into the letter.

To draw a letter in the center of a box with a color and a font:

Put the letter into a string.

Draw the string in the box with the color and the font and “center”.

To create a list of brightness blocks from a picture:

Center the picture on the screen.

Draw the picture.

Put the picture’s left-top into a spot.

Loop.

Allocate memory for a block. Append the block to the list.

Put the spot and the spot plus 12 pixels into the block’s box.

Put the block’s average brightness into the block’s brightness.

Add 12 pixels to the spot's left. If the spot's left is less than the picture's right, repeat.  
Put the picture's left into the spot's left.  
Add 12 pixels to the spot's top. If the spot's top is less than the picture's bottom, repeat.

To put a block's average brightness into a brightness:

Put the block's left-top into a spot.

Loop.

Get a color given the spot.

Add the color's brightness to a total brightness. Add 1 to a count.

Add 1 pixel to the spot's left. If the spot is in the block's box, repeat.

Put the block's left into the spot's left.

Add 1 pixel to the spot's top. If the spot is in the block's box, repeat.

Put the total divided by the count into the brightness.

To fetch a picture from a path:

Read the path into the picture.

Resize the picture to 8 inches by 8 inches.

If you prefer a different language, just think of the above as pseudo-code.